Chapter 15
A Systematic Review of Distributed Software Development
Problems and Solutions

Miguel Jiménez
Alhambra-Eidos, Spain

Mario Piattini
University of Castilla-La Mancha, Spain

Aurora Vizcaíno
University of Castilla-La Mancha, Spain

ABSTRACT

In last years, software development activity tends to be decentralized, thus expanding greater development efforts towards more attractive zones for organizations. The type of development in which the team members are distributed in remote sites is called distributed software development (DSD). The main advantage of this practice is mainly that of having a greater availability of human resources in decentralized zones with less cost. On the other hand, organizations must face some disadvantages due to the distance that separates the development teams related to project organization, project control and product quality. Coordination becomes more difficult as the software components are sourced from different places, and new processes and tools are needed. This chapter presents a systematic review of the literature related to the problems of DSD with the purpose of obtaining a vision about the solutions proposed up to the present day.

1 INTRODUCTION

Nowadays, software industry tends to relocate their production units to decentralized zones with greater availability of skilled workforce, taking advantage of politic and economic factors (Aspray, et al., 2006). The objective consists of optimizing resources in order to develop higher quality products at a lower cost than is in co-located developments. In this sense, Software Factories (Greenfield, et al., 2004)

DOI: 10.4018/978-1-60566-731-7.ch015

is an approach that, automate parts of software development by imitating industrial processes originally linked to more traditional sectors such as those of the automobile and aviation, decentralize production units, and promote the reusability of architectures, knowledge and components.

Distributed Software Development (DSD) allows the team members to be located in various remote sites during the software lifecycle, thus making up a network of distant sub-teams. In this context the traditional face-to-face meetings are no longer common and interaction between members requires the use of technology to facilitate communication and coordination.

The distance between the different teams can vary from a few meters (when the teams work in adjacent buildings) to different continents (Prikladnicki, et al., 2003). The situation in which the teams are distributed beyond the limits of a nation is called Global Software Development (GSD). This kind of scenario is interesting for several reasons (Herbsleb and Moitra, 2001), mainly because it enables organizations to abstract themselves from geographical distance, whilst having qualified human resources and minimizing cost (Werner, et al., 2001), increasing their market area by producing software for remote clients and obtaining a longer workday by taking advantage of time differences (Ebert & De Neve, 2001). On the other hand we must confront a number of problems (Layman, et al., 2006), caused mainly by distance and time and cultural differences (Krishna, et al., 2004), which depend largely on the specific characteristics of each organization.

In this context, "offshoring" refers to the transfer of an organizational function to another country, usually where human resources are cheaper. We speak about "nearshoring" when jobs are transferred to geographically closer countries, thus avoiding cultural and time differences between members and saving travel and communication costs. Outsourcing is a way to contract an external organization, independently of its location, instead of developing in-house (McConnell, 1996).

The aforementioned development practices have as a common factor the problems arising from distance that directly affect the processes of communication as well as coordination and control activities (Damian, et al., 2003). In these environments, communication is less fluid than in colocalized development groups, as a consequence, problems related to coordination, collaboration or group awareness appear which negatively affect productivity and, consequently, software quality. All these factors influence the way in which software is defined, built, tested and delivered to customers, thus affecting the corresponding stages of the software life cycle.

In order to mitigate these effects and with the aim of achieving higher levels of productivity, organizations require facilities to support collaboration, coordination and communication among developers through new technologies, processes and methods (Damian and Lanubile, 2004). Iterative approaches are commonly used in contrast to traditional waterfall or sequential methods, but they become more difficult to use consistently when teams are geographically distributed (Cusumano, 2008).

This work presents a systematic review of the literature dealing with efforts related to DSD and GSD with the purpose of discovering the aspects upon which researchers have focused until this moment. The objective is to identify, evaluate, interpret and synthesize most of the important studies on the subject, by conducting a rigorous and objective review of literature which will allow us to analyze the issues and the solutions contributed up to the present about de-located development with the aim of obtaining information with a high scientific and practical value through a rigorous systematic method.